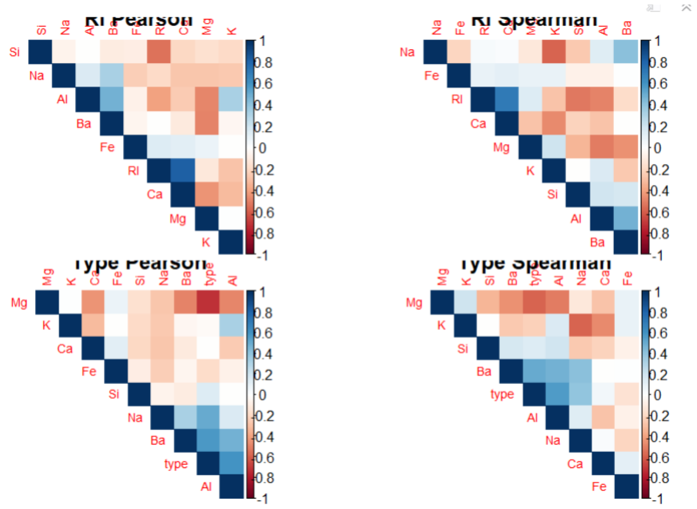
Introduction to Data analytics

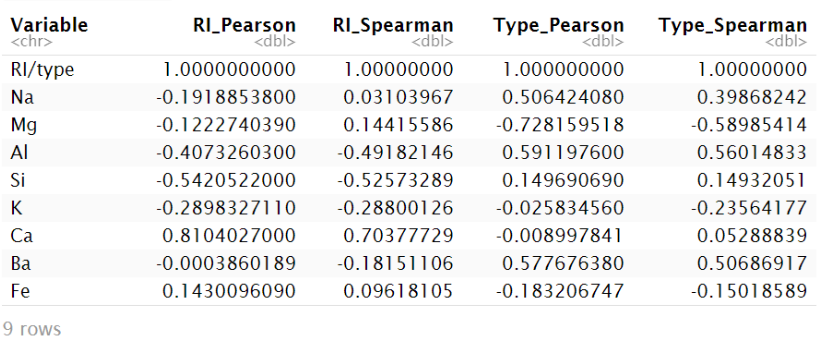
Glass

The glass dataset was created in 1987 by the USA Forensic Science Service and was used in a comparison test for the rule-based nearest-neighbour algorithm, BEAGLE. The dataset contains the results to a test determining whether the glass was a type of “float” glass or not. (https://archive.ics.uci.edu/dataset/42/glass+identification)

The dataset is made up of 11 variables and 214 observations. The first variable is the unique identifier of the pieces of glass tested and is denoted as “ID_number” ranging from 1 to 214 taking an integer as its data type. The next variable is the first response variable of the dataset, it records the reflective index “RI”, of the piece of glass. Next there are 8 variables denoting the number of units of different elements that appear in the glass, they are “Na” sodium, “Mg” magnesium, “Al” aluminium, “Si” silicon, “K” potassium, “Ca” calcium, “Ba” barium and “Fe” iron, all taking a numerical data type. Finally, we have a factor variable which holds the predicted type of glass, ranging in 6 levels, types 1, 2, 3, 5, 6 and 7.

The first analytical technique we can use to gain a general understanding of how the variables relate to each other is using a correlation test and PCA test. We tried two different methods of correlation type to verify that the calculations were correct. The two method we used were Pearson and Spearman methods. The results from this test were given in a heatmap for simplicity and ease of understanding by the audience.

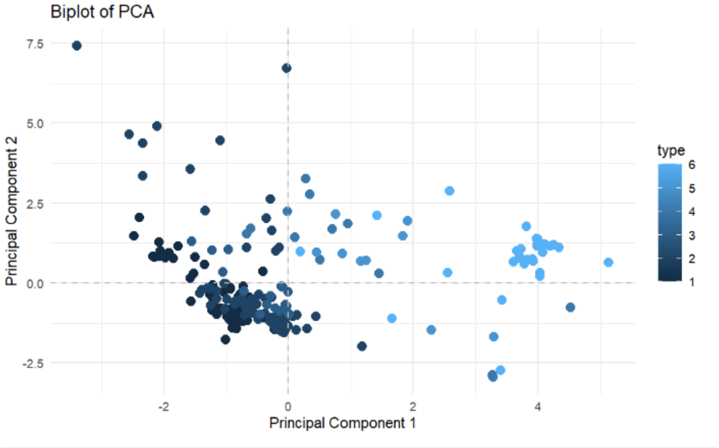




Variable <chr>	RI_Pearson <dbl>	RI_Spearman <dbl>	Type_Pearson <dbl>	Type_Spearman <dbl>
RI/type	1.0000000000	1.00000000	1.000000000	1.00000000
Na	-0.1918853800	0.03103967	0.506424080	0.39868242
Mg	-0.1222740390	0.14415586	-0.728159518	-0.58985414
Al	-0.4073260300	-0.49182146	0.591197600	0.56014833
Si	-0.5420522000	-0.52573289	0.149690690	0.14932051
K	-0.2898327110	-0.28800126	-0.025834560	-0.23564177
Ca	0.8104027000	0.70377729	-0.008997841	0.05288839
Ba	-0.0003860189	-0.18151106	0.577676380	0.50686917
Fe	0.1430096090	0.09618105	-0.183206747	-0.15018589

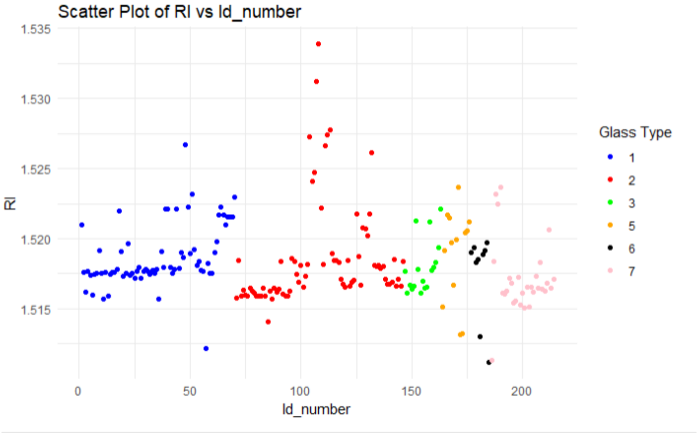
9 rows

The table above shows the correlation between explanatory variables and the two response variables in RI and Type. From this we see that the in both cases both methods gave similar results, adding to the validity of the test. For RI only 1 variable showed strong correlation, calcium with 0.8104026963, the next highest correlation was Silicon with -0.5420521997, this however is not as strong of a correlation. When correlating with type, similarly only 1 variable had a strong correlation, magnesium with -0.728159518 the next largest correlation again does not have a strong correlation, aluminium with 0.591197598. The completing dimension reduction through a PCA test, we found that 53.4% of the prediction could be found through the first 2 principal components. When plotting this we achieve:

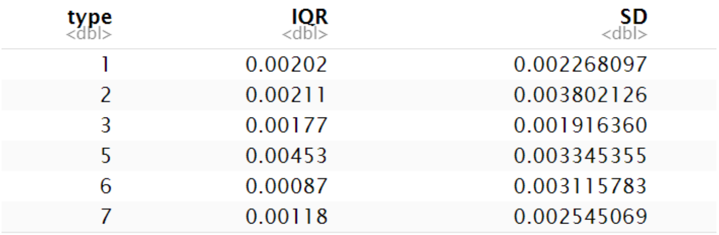


This shows a correlation between the lower types of glass and gaining more variation through to the higher types of glass.

The lack of correlation with RI was further shown when creating a scatter plot.



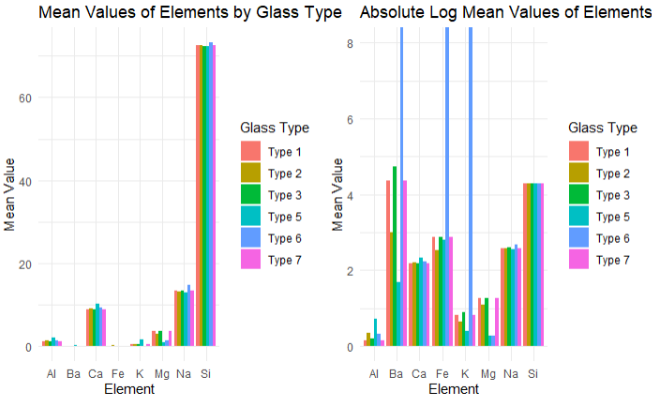
This graph shows all the pieces of glass individually, and what there corresponding reflective indexes, they are then grouped by the type of glass predicted. From this we don’t see any clear patterns between types and reflective indexes, it seems that any type of glass can have a wide range of refractive indexes. However, we can look at the spread of the analytically by comparing the standard deviations and inter-quartile ranges of each group.



type <dbl>	IQR <dbl>	SD <dbl>
1	0.00202	0.002268097
2	0.00211	0.003802126
3	0.00177	0.001916360
5	0.00453	0.003345355
6	0.00087	0.003115783
7	0.00118	0.002545069

Where comparing these values we see that types 2 and 5 have the most amount of variation in refractive index, 5 having the largest inter-quartile range and 2 having the largest standard deviation. We also see that type 6 has the smallest IQR whereas 3 has the smallest standard deviation thus they are denoted as with the least variation in reflective index.

The lack of correlation is also shown when comparing with type. As type is a factor variable to most clear and concise way of graphically translating the information is through a bar chart, with the factor type on the x axis and the mean values on the y axis.

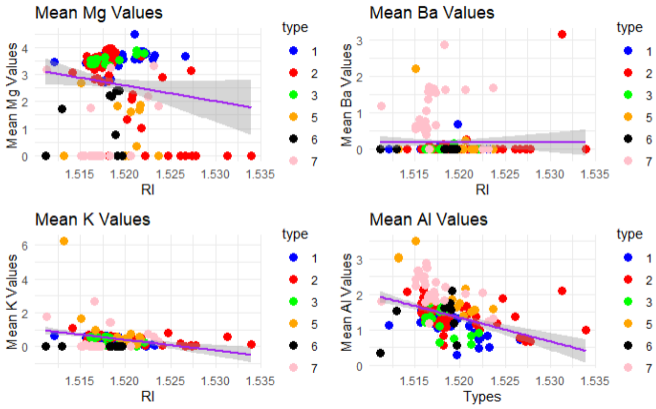


When looking at the graph on the left me see that because of the large amounts of Silicon in every piece of glass, the scale of the y axis is off. Smaller increments in the bars will be less visible thus could be overlooked by the audience. To combat this, I have used a log transform to scale the values down to similar levels but keep the ratios and proportions intact. This resulted in another issue however as the values from the log mean for the potassium, iron and barium columns gained negative values, as negative quantities do not make intuitive sense, we must take the absolute value, giving the graph on the right.

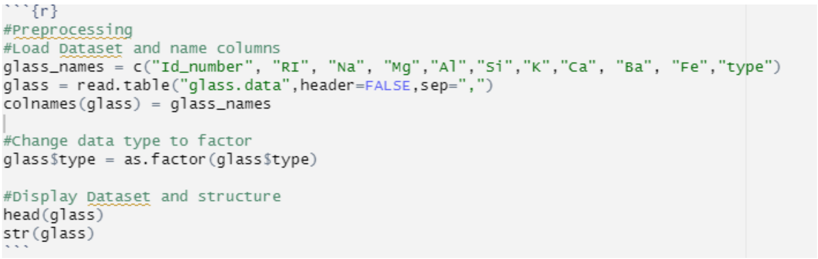
The first point of interest in this bar chart is the consistency of the silicon, sodium and calcium values. For all 6 types of glass the sodium levels do not significantly change, additionally this is also true for sodium and calcium except for a of a minor increases in type 6 glass. This implies that for all 3 elements do not correlate well with glass and are bad predictors. Iron similarly does not have high variation, 4 types being the same but types 2 having slightly less, and types 6 have drastically more. The elements with more variation are Barium, librar, potassium and magnesium the amount all vary much more between types, giving evidence to the fact that they have a higher correlation and would be better predictors.

Looking back at the table of correlations, we found that this is true Barium, aluminium and magnesium all had the largest correlations with type. From these two pieces of evidence, we can conclude that they are the 3 best predictors. If we used all 3 in a PCA test we could gather 67.5% of the variation in variables.

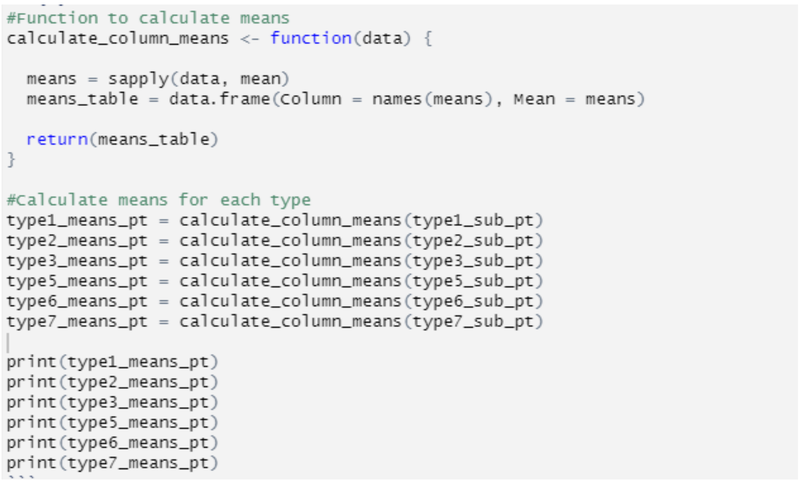
If we plot these 4 predictors, we can find trends with the number of metals and the type of glass.



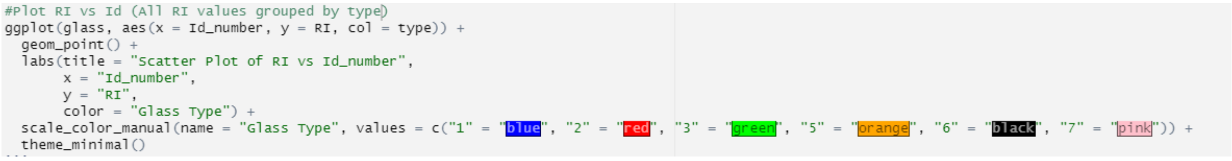
These plots suggest that if there are high amount of Potassium, Barium or Aluminium we can be certain that it would be type 5 glass, it also implies that if we record low amount of magnesium, we can be certain it is type 5 or 6.



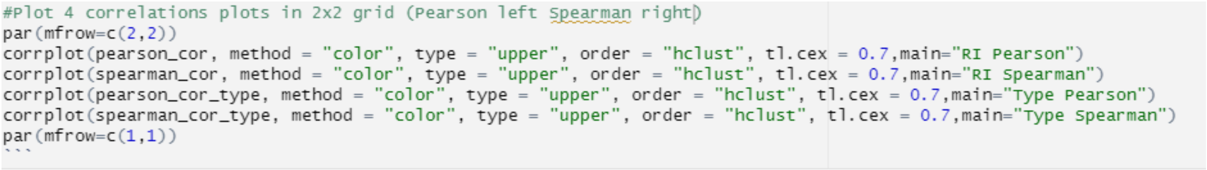
~~~
```{r}
#Preprocessing
#Load Dataset and name columns
glass_names = c("Id_number", "RI", "Na", "Mg","Al","Si","K","Ca", "Ba", "Fe","type")
glass = read.table("glass.data",header=FALSE,sep=",")
colnames(glass) = glass_names

#Change data type to factor
glass$type = as.factor(glass$type)

#Display Dataset and structure
head(glass)
str(glass)
```
~~~



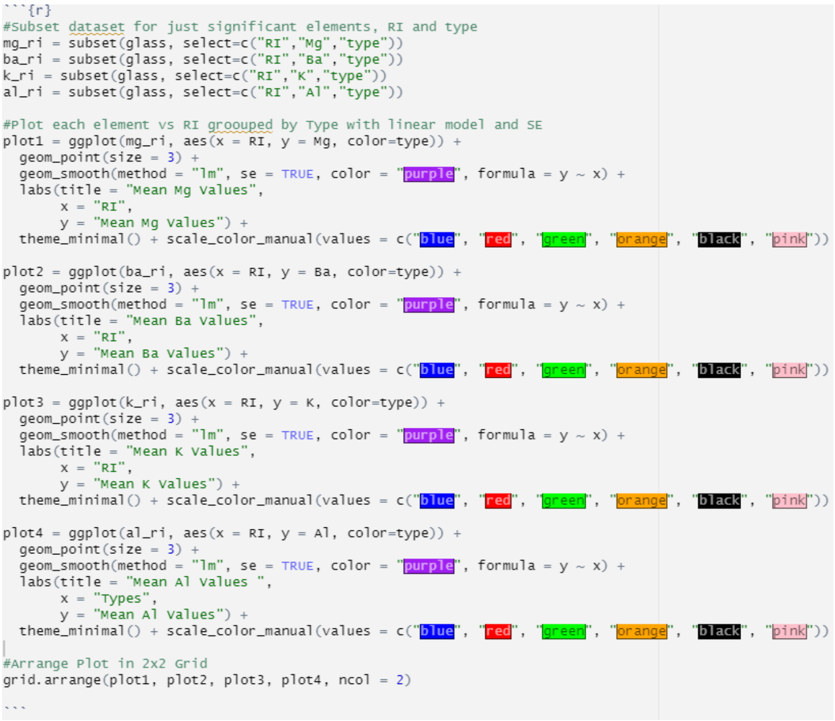
~~~
#Function to calculate means
calculate_column_means <- function(data) {

  means = sapply(data, mean)
  means_table = data.frame(Column = names(means), Mean = means)

  return(means_table)
}

#Calculate means for each type
type1_means_pt = calculate_column_means(type1_sub_pt)
type2_means_pt = calculate_column_means(type2_sub_pt)
type3_means_pt = calculate_column_means(type3_sub_pt)
type5_means_pt = calculate_column_means(type5_sub_pt)
type6_means_pt = calculate_column_means(type6_sub_pt)
type7_means_pt = calculate_column_means(type7_sub_pt)

print(type1_means_pt)
print(type2_means_pt)
print(type3_means_pt)
print(type5_means_pt)
print(type6_means_pt)
print(type7_means_pt)
```
~~~



````
#Plot RI vs Id (All RI values grouped by type)
ggplot(glass, aes(x = Id_number, y = RI, col = type)) +
  geom_point() +
  labs(title = "Scatter Plot of RI vs Id_number",
       x = "Id_number",
       y = "RI",
       color = "Glass Type") +
  scale_color_manual(name = "Glass Type", values = c("1" = "blue", "2" = "red", "3" = "green", "5" = "orange", "6" = "black", "7" = "pink")) +
  theme_minimal()
````



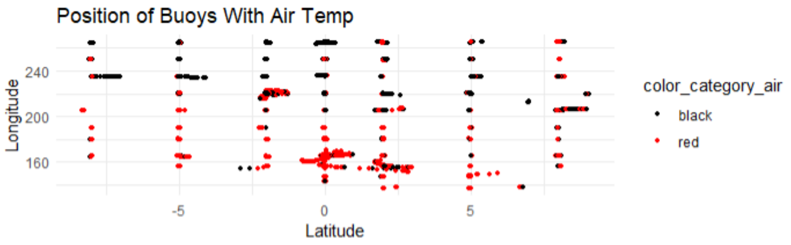
~~~
#Plot 4 correlations plots in 2x2 grid (Pearson left Spearman right)
par(mfrow=c(2,2))
corrplot(pearson_cor, method = "color", type = "upper", order = "hclust", tl.cex = 0.7,main="RI Pearson")
corrplot(spearman_cor, method = "color", type = "upper", order = "hclust", tl.cex = 0.7,main="RI Spearman")
corrplot(pearson_cor_type, method = "color", type = "upper", order = "hclust", tl.cex = 0.7,main="Type Pearson")
corrplot(spearman_cor_type, method = "color", type = "upper", order = "hclust", tl.cex = 0.7,main="Type Spearman")
par(mfrow=c(1,1))
```
~~~

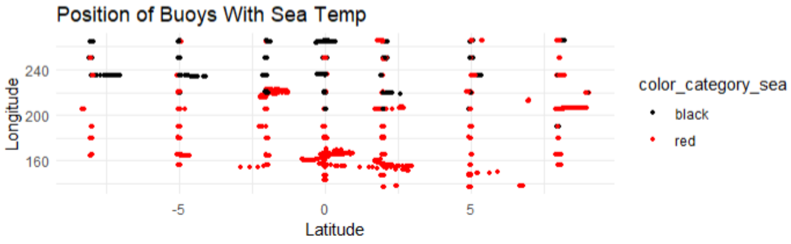


~~~
```{r}
#Subset dataset for just significant elements, RI and type
mg_ri = subset(glass, select=c("RI","Mg","type"))
ba_ri = subset(glass, select=c("RI","Ba","type"))
k_ri = subset(glass, select=c("RI","K","type"))
al_ri = subset(glass, select=c("RI","Al","type"))

#Plot each element vs RI groouped by Type with linear model and SE
plot1 = ggplot(mg_ri, aes(x = RI, y = Mg, color=type)) +
  geom_point(size = 3) +
  geom_smooth(method = "lm", se = TRUE, color = "purple", formula = y ~ x) +
  labs(title = "Mean Mg Values",
       x = "RI",
       y = "Mean Mg Values") +
  theme_minimal() + scale_color_manual(values = c("blue", "red", "green", "orange", "black", "pink"))

plot2 = ggplot(ba_ri, aes(x = RI, y = Ba, color=type)) +
  geom_point(size = 3) +
  geom_smooth(method = "lm", se = TRUE, color = "purple", formula = y ~ x) +
  labs(title = "Mean Ba Values",
       x = "RI",
       y = "Mean Ba Values") +
  theme_minimal() + scale_color_manual(values = c("blue", "red", "green", "orange", "black", "pink"))

plot3 = ggplot(k_ri, aes(x = RI, y = K, color=type)) +
  geom_point(size = 3) +
  geom_smooth(method = "lm", se = TRUE, color = "purple", formula = y ~ x) +
  labs(title = "Mean K Values",
       x = "RI",
       y = "Mean K Values") +
  theme_minimal() + scale_color_manual(values = c("blue", "red", "green", "orange", "black", "pink"))

plot4 = ggplot(al_ri, aes(x = RI, y = Al, color=type)) +
  geom_point(size = 3) +
  geom_smooth(method = "lm", se = TRUE, color = "purple", formula = y ~ x) +
  labs(title = "Mean Al Values ",
       x = "Types",
       y = "Mean Al Values") +
  theme_minimal() + scale_color_manual(values = c("blue", "red", "green", "orange", "black", "pink"))

#Arrange Plot in 2x2 Grid
grid.arrange(plot1, plot2, plot3, plot4, ncol = 2)

```
~~~

El Nino

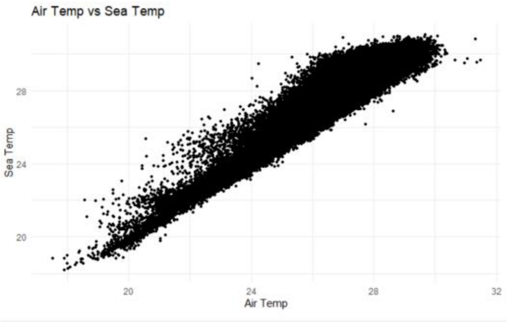
The El Nino dataset records oceanographic and surface meteorological data recorded from a series of boys positioned throughout the equatorial Pacific Ocean. The data was extracted by the Tropical Atmosphere Ocean (TOA) array which was developed by the international Tropical Global Atmosphere (TOGA) programme. The array consists of 70 moored buoys measuring oceanographic and meteorological variables critical for improved detection and prediction of climate variation. This dataset as 12 variables formulating the date at which the reading took place the latitude and longitude cartesian coordinates of the buoy and finally wind speeds, humidity and air and sea temperatures at that given position. Some pre-processing formatting had to be completed with this dataset however, as the column for humidity was empty. As this empty column had no use it was removed entirely. A date column was constructed from the year month and day values, collecting them in the date data type form of YYYY-MM-DD. The other difference to the glass dataset was the longitude and latitude columns, as these did not hold recorded values, they represented the cartesian coordinates of the position of the buoy. The allow this to be represented correctly, we added 360 to the longitude to all values below 0. This allowed the graph to 'loop' like a map, allowing the points on the left-hand side of the graph to be next to the points on the right.



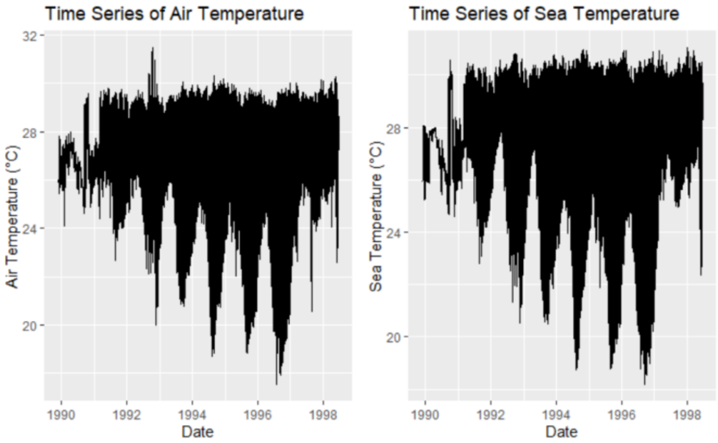


When mapping the buoys in this manner we gain the figure above. As you can see the buoys follow a structured layout of longitudinal stripes ranging from 136 to 265, evenly spaced in the latitude direction, with gaps of around 2.5 between the stripes. The colour of the buoys represents the temperature in the air and the sea recorded at each position, points coloured black had a temperature of less than 27 degrees Celsius and points labelled in red had a temperature of above 27 degrees Celsius, this value was selected to be a good boundary as it is the median value of the air temperature column. The air temperature records have a minimum of 17.54 and the maximum of 31.48, while in sea temperature, the values range from 18.19 to 31.04. We would expect the sea temperature to have a smaller interquartile range due to the larger heat capacity of water.

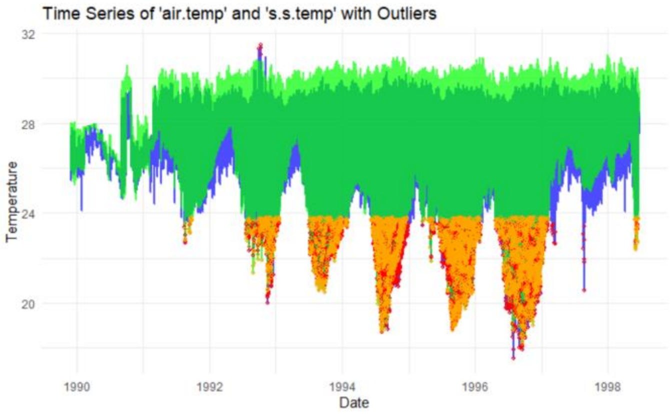
Through further inspection we can see that the temperature in air and sea have a very similar pattern of cold values vs hot values. We see in both cases that the buoys in the northern half, above 200 longitude, have a significantly colder temperature. The air temperature varies slightly more then the sea temperature, as air takes less time than the denser water to heat up thus, we see more odd buoys in the southern half of the map that record less than 27 degrees Celsius. This suggests that air temperature and sea temperature have a strong positive correlation. When plotting these two variables this assumption is proven correct.



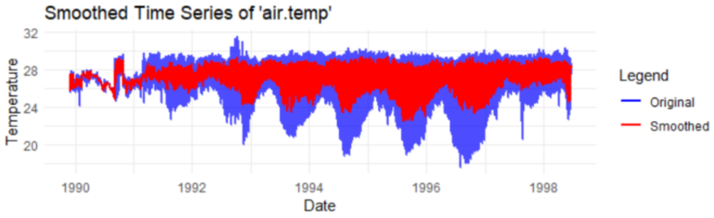
The other variable of interest is the date variable, constructed from the year month and day values. As this is a time variable the best way to represent this data is using a time series.

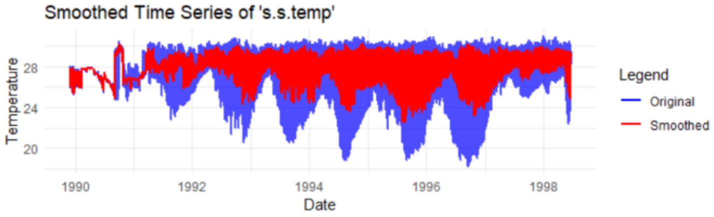


Again, due to the strong positive correlation, the time series of the 2 temperatures have very similar patterns. We see a harmonic pattern moving from the minimum temperature to the maximum temperature every 6 months. As the time series includes a lot of data points, to looks very thick on the graph this could mean that more intricate patterns that live in the data might not be visible. To check this, we need to look for outliers in the data to remove, this will reduce noise in the data and will allow us to see more intricate details of the data.

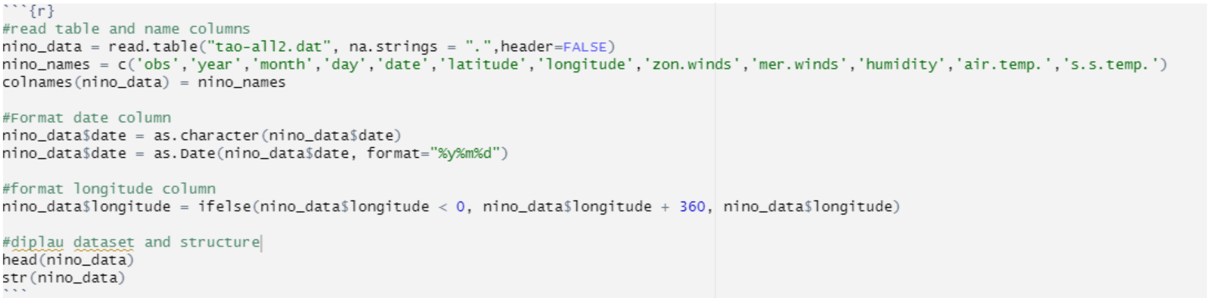


In the graph above, we see the time series for air temperature in blue with the outliers highlighted red, and the time series for sea temperature in green with outlies highlighted orange. As expected, both time series have a lot of outliers in the data, nearly all the extreme low values are given as outliers. This is suggesting that the time series could be improve using a median smoothing.

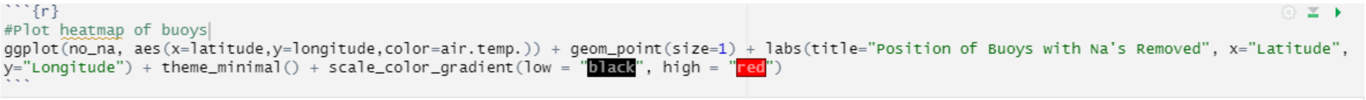




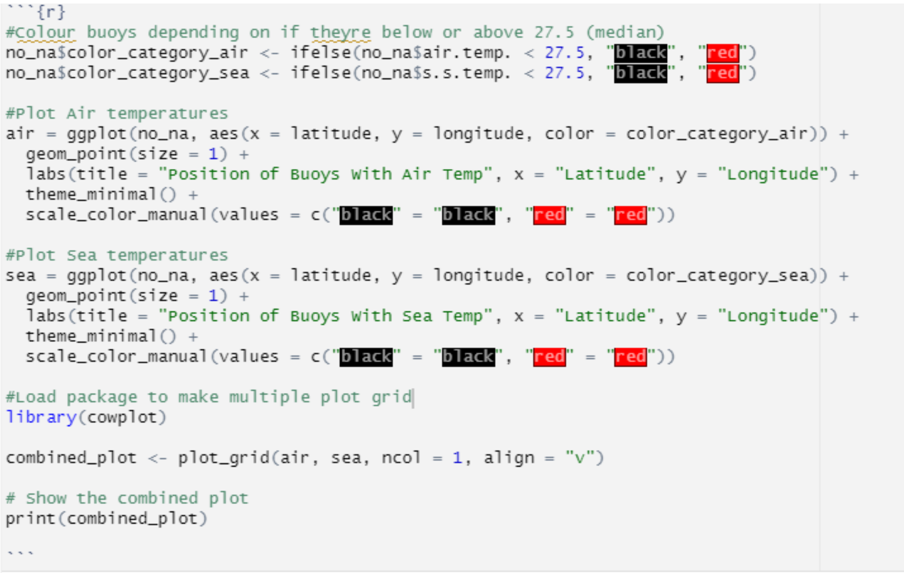
As a lot of the redundant data or noise was removed from the graphs, we can see that the temperature in truth has a lot less variation than the previous graphs suggested. We find that the smoothed time series gives a maximum value of 29.42 and a minimum value of 22.44, giving a range of 6.98 in air temperature and in sea temperature we gain a range of 7.77 degrees, from 22.57 to 30.34 degrees Celsius. We see that the harmonic pattern persists repeating the same pattern on a yearly cycle. Having a peak low temperature in January, slowly rising to a peak high in May and June, moving back down to a new low in august to October, and staying that low to repeat the pattern in January again.



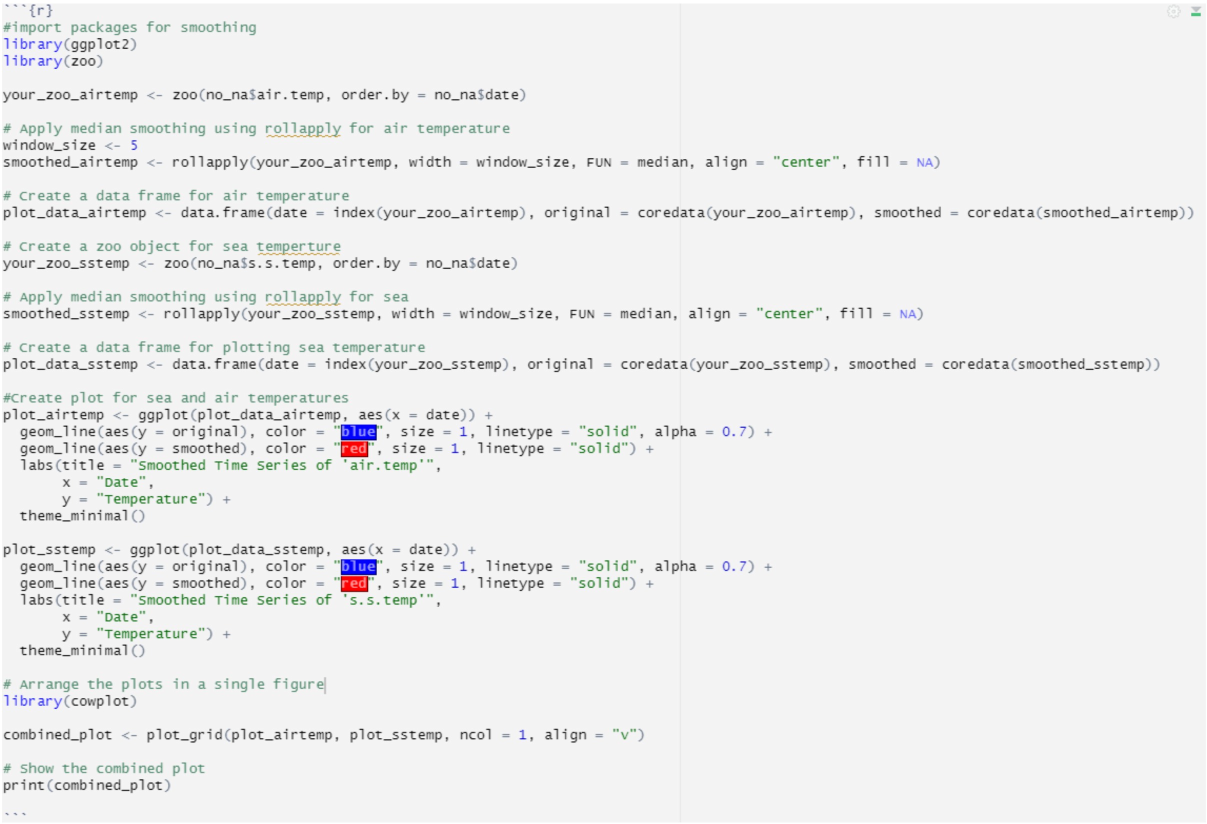
~~~
```{r}
#read table and name columns
nino_data = read.table("tao-all2.dat", na.strings = ".",header=FALSE)
nino_names = c('obs','year','month','day','date','latitude','longitude','zon.winds','mer.winds','humidity','air.temp.','s.s.temp.')
colnames(nino_data) = nino_names

#Format date column
nino_data$date = as.character(nino_data$date)
nino_data$date = as.Date(nino_data$date, format="%y%m%d")

#format longitude column
nino_data$longitude = ifelse(nino_data$longitude < 0, nino_data$longitude + 360, nino_data$longitude)

#diplau dataset and structure
head(nino_data)
str(nino_data)
```
~~~



~~~
```{r}
#Plot heatmap of buoys
ggplot(no_na, aes(x=latitude,y=longitude,color=air.temp.)) + geom_point(size=1) + labs(title="Position of Buoys with Na's Removed", x="Latitude",
y="Longitude") + theme_minimal() + scale_color_gradient(low = "black", high = "red")
```
~~~



~~~
```{r}
#Colour buoys depending on if theyre below or above 27.5 (median)
no_na$color_category_air <- ifelse(no_na$air.temp. < 27.5, "black", "red")
no_na$color_category_sea <- ifelse(no_na$s.s.temp. < 27.5, "black", "red")

#Plot Air temperatures
air = ggplot(no_na, aes(x = latitude, y = longitude, color = color_category_air)) +
  geom_point(size = 1) +
  labs(title = "Position of Buoys With Air Temp", x = "Latitude", y = "Longitude") +
  theme_minimal() +
  scale_color_manual(values = c("black" = "black", "red" = "red"))

#Plot Sea temperatures
sea = ggplot(no_na, aes(x = latitude, y = longitude, color = color_category_sea)) +
  geom_point(size = 1) +
  labs(title = "Position of Buoys With Sea Temp", x = "Latitude", y = "Longitude") +
  theme_minimal() +
  scale_color_manual(values = c("black" = "black", "red" = "red"))

#Load package to make multiple plot grid
library(cowplot)

combined_plot <- plot_grid(air, sea, ncol = 1, align = "v")

# Show the combined plot
print(combined_plot)

```
~~~

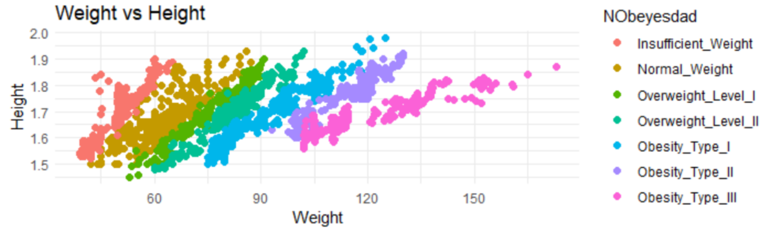


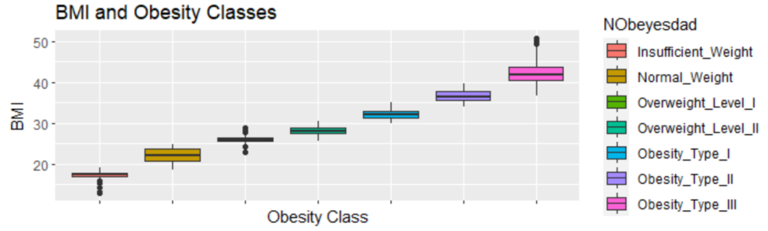
~~~
```{r}
#import packages for smoothing
library(ggplot2)
library(zoo)

your_zoo_airtemp <- zoo(no_na$air.temp, order.by = no_na$date)

# Apply median smoothing using rollapply for air temperature
window_size <- 5
smoothed_airtemp <- rollapply(your_zoo_airtemp, width = window_size, FUN = median, align = "center", fill = NA)

# Create a data frame for air temperature
plot_data_airtemp <- data.frame(date = index(your_zoo_airtemp), original = coredata(your_zoo_airtemp), smoothed = coredata(smoothed_airtemp))

# Create a zoo object for sea temperture
your_zoo_sstemp <- zoo(no_na$s.s.temp, order.by = no_na$date)

# Apply median smoothing using rollapply for sea
smoothed_sstemp <- rollapply(your_zoo_sstemp, width = window_size, FUN = median, align = "center", fill = NA)

# Create a data frame for plotting sea temperature
plot_data_sstemp <- data.frame(date = index(your_zoo_sstemp), original = coredata(your_zoo_sstemp), smoothed = coredata(smoothed_sstemp))

#Create plot for sea and air temperatures
plot_airtemp <- ggplot(plot_data_airtemp, aes(x = date)) +
  geom_line(aes(y = original), color = "blue", size = 1, linetype = "solid", alpha = 0.7) +
  geom_line(aes(y = smoothed), color = "red", size = 1, linetype = "solid") +
  labs(title = "Smoothed Time Series of 'air.temp'",
       x = "Date",
       y = "Temperature") +
  theme_minimal()

plot_sstemp <- ggplot(plot_data_sstemp, aes(x = date)) +
  geom_line(aes(y = original), color = "blue", size = 1, linetype = "solid", alpha = 0.7) +
  geom_line(aes(y = smoothed), color = "red", size = 1, linetype = "solid") +
  labs(title = "Smoothed Time Series of 's.s.temp'",
       x = "Date",
       y = "Temperature") +
  theme_minimal()

# Arrange the plots in a single figure
library(cowplot)

combined_plot <- plot_grid(plot_airtemp, plot_sstemp, ncol = 1, align = "v")

# Show the combined plot
print(combined_plot)

```
~~~

Obesity

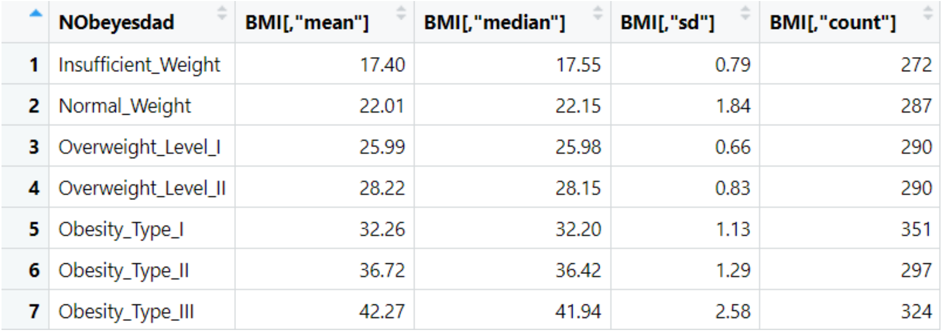
The obesity dataset is an estimation of obesity levels based on eating habits and physical conditions. The dataset includes data collected on individuals from Mexico, Peru and Colombia. It records general figures like age, height and weight, then some health and lifestyle related questions such as 'do you smoke?', 'do you eat high caloric foods?', 'do you usually eat vegetables?' or 'how you do travel to work?'. Finally having a response variable in obesity level, having 7 different levels 'Insufficient weight', 'Normal Weight', 'Overweight Level I', 'Overweight Level II' 'Obesity Type I', 'Obesity Type II' and 'Obesity Type III'. (https://archive.ics.uci.edu/dataset/544/estimation+of+obesity+levels+based+on+eating+habits+and+physical+condition)

Clearly the variable for height is going to be a strong correlator for the weight class of the individual however taking this variable as the sole response variable could be misleading as the height of the individual puts into context their weight. As a result, we will calculate the BMI to adjust the response variable to be standardised in an understandable way. From more in-depth studies, we find that in reality BMI is not a good predictor of health or obesity, however when plotting these BMI scores we see a almost perfect strong positive correlation with our individuals and their data.





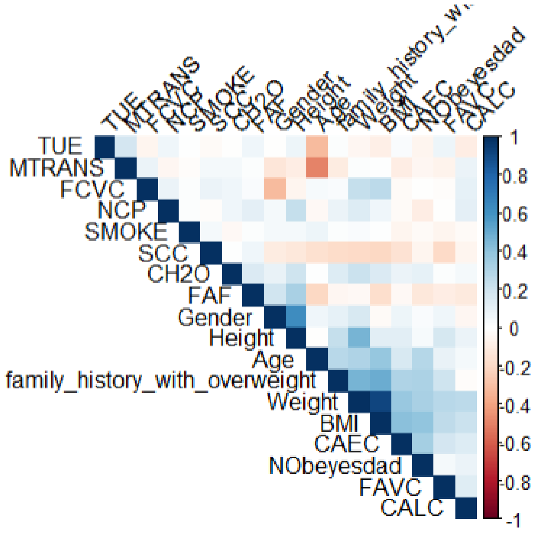
As you can see this created variable BMI can be used as a good continuous estimator for the discrete categories, allowing both a continuous response and a categorical response able to be analyse. When looking at the summary statistics for each group we find this table.



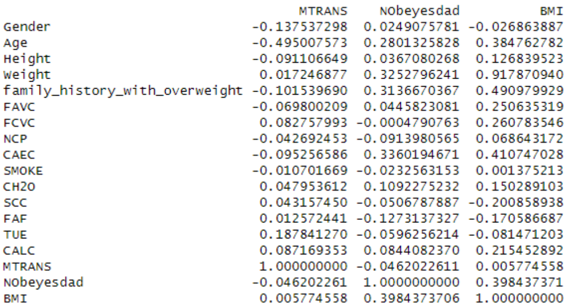
	NObeyesdad	BMI[,"mean"]	BMI[,"median"]	BMI[,"sd"]	BMI[,"count"]
1	Insufficient_Weight	17.40	17.55	0.79	272
2	Normal_Weight	22.01	22.15	1.84	287
3	Overweight_Level_I	25.99	25.98	0.66	290
4	Overweight_Level_II	28.22	28.15	0.83	290
5	Obesity_Type_I	32.26	32.20	1.13	351
6	Obesity_Type_II	36.72	36.42	1.29	297
7	Obesity_Type_III	42.27	41.94	2.58	324

From this we can see that the variation in each group is relatively small with a standard deviation of around 0.79 to 1.84 in all cases apart from the most obese category, 'Obesity_Type_III' with a standard deviation of 2.58. Each group has an average difference of 4.48 and 272 to 351 people in each group.

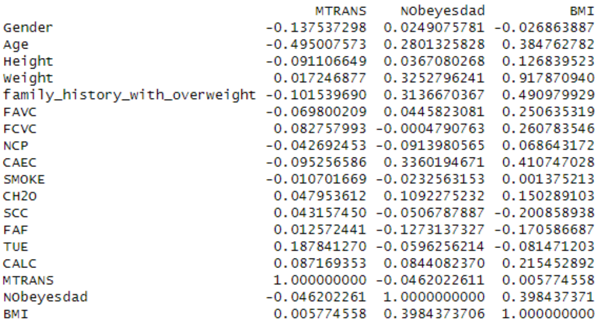
To gain a general understanding of how the variables relate to each other, a correlation test can be performed however, in our case we have the issue of a lot of our variables being categorical data types, which cannot be used in a correlation test first, we must convert all variables to the numeric data type to allow the tests implementation. A correlation heatmap is then constructed to relay this information.



The Pearson correlation test results and are on the left whereas the Spearman correlation test results are on the right:



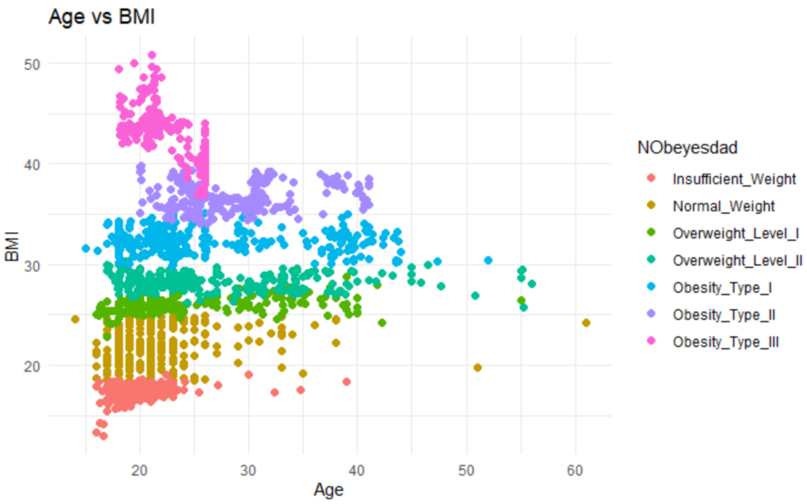
```
                                  MTRANS   NObeyesdad          BMI
Gender                      -0.137537298 0.0249075781 -0.026863887
Age                         -0.495007573 0.2801325828  0.384762782
Height                      -0.091106649 0.0367080268  0.126839523
Weight                       0.017246877 0.3252796241  0.917870940
family_history_with_overweight -0.101539690 0.3136670367 0.490979929
FAVC                        -0.069800209 0.0445823081  0.250635319
FCVC                         0.082757993 -0.0004790763 0.260783546
NCP                         -0.042692453 -0.0913980565 0.068643172
CAEC                        -0.095256586 0.3360194671  0.410747028
SMOKE                       -0.010701669 -0.0232563153 0.001375213
CH2O                         0.047953612 0.1092275232  0.150289103
SCC                          0.043157450 -0.0506787887 -0.200858938
FAF                          0.012572441 -0.1273137327 -0.170586687
TUE                          0.187841270 -0.0596256214 -0.081471203
CALC                         0.087169353 0.0844082370  0.215452892
MTRANS                       1.000000000 -0.0462022611 0.005774558
NObeyesdad                  -0.046202261 1.0000000000  0.398437371
BMI                          0.005774558 0.3984373706  1.000000000
```



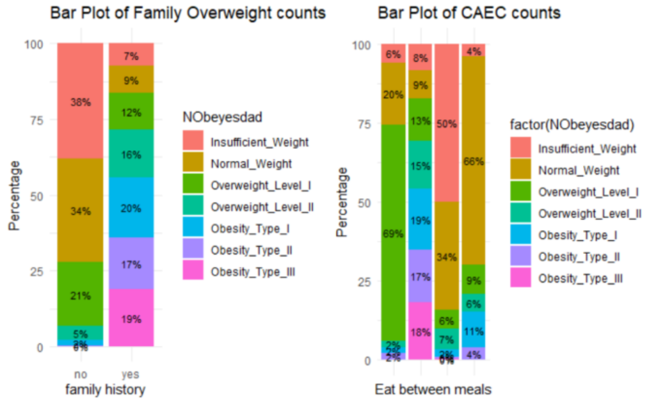
```
                                  MTRANS   NObeyesdad          BMI
Gender                      -0.137537298 0.0249075781 -0.026863887
Age                         -0.495007573 0.2801325828  0.384762782
Height                      -0.091106649 0.0367080268  0.126839523
Weight                       0.017246877 0.3252796241  0.917870940
family_history_with_overweight -0.101539690 0.3136670367 0.490979929
FAVC                        -0.069800209 0.0445823081  0.250635319
FCVC                         0.082757993 -0.0004790763 0.260783546
NCP                         -0.042692453 -0.0913980565 0.068643172
CAEC                        -0.095256586 0.3360194671  0.410747028
SMOKE                       -0.010701669 -0.0232563153 0.001375213
CH2O                         0.047953612 0.1092275232  0.150289103
SCC                          0.043157450 -0.0506787887 -0.200858938
FAF                          0.012572441 -0.1273137327 -0.170586687
TUE                          0.187841270 -0.0596256214 -0.081471203
CALC                         0.087169353 0.0844082370  0.215452892
MTRANS                       1.000000000 -0.0462022611 0.005774558
NObeyesdad                  -0.046202261 1.0000000000  0.398437371
BMI                          0.005774558 0.3984373706  1.000000000
```

In the Pearson test we find that the value with the highest correlation with Obesity categories are CAEC (Do you eat between meals), family_history_with_overweight and Age. The variables with the least correlation were FCVC (Do you usually eat vegetables?), smoke (Do you smoke?), and Gender. When looking at BMI the variables with the highest correlation were, family_history_with_overweight, CAEC (Do you eat between meals?) and Age. The variables with the lest correlation with BMI were smoke (Do you smoke?) MTrans (How do you travel to work?) and NCP (How many main meals do you have a day?). When examining each list we have that the 3 variables that are most correlated and the same for each variable, however when comparing the least correlated some variation is observed, smoke appears in both lists however MTrans and NCP appear in BMI and FCVC and gender appear in the categories. When evaluating the Spearman test, we see that the results are the same in all counts but instead of NCP as the least correlated it selected TUE (How long do you use tech for?). It is of note the general low correlation between individual variables and the response variable, that highest value given was 0.498 which would not be regarded as strong correlation. This is implying that not one variable is the reason for being in an obesity class, instead it is a combination of many factors or variables.

The next step in understand the dataset is to visually examine the important variables to see how they affect the response variables. Family_history_with_overweight and CAEC are both categorical variables, thus the best way to display their data is through a bar chart. Age on the other hand is continuous, so we should plot it in a scatter plot, plotting against BMI. Unfortunately, when trying to group these plots in the same figure the plots become squished that unreadable thus had to be copied in individually.

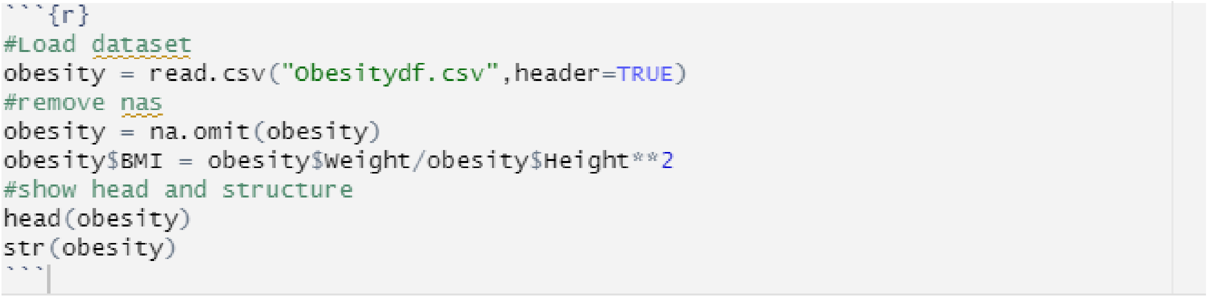


Firstly, we are looking at the scatter plot between Age and BMI, we see that we tend to see that the higher BMI values are only populating the lower age groups with the top two brackets Obesity_Type_II and Obesity_Type_III peaking at ages 41 and 26 respectively. It seems that in the younger half there seems to be an even split between the groups, after around 25 ages of year the probability of an individual being in the obese class increases with very few counts of normal weights. As we move to the older participants, we find that most are in the overweight_II category with one obesity_type_I and 2 normal_weights, This however cannot be taken with much confidence as the sample size in older individuals is very small.

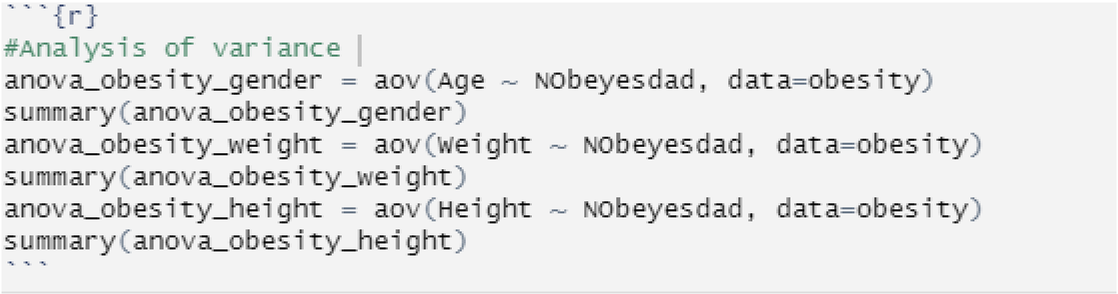


The most important categorical variable was family_history_with_overweight, shown on the left-hand side of the figure. We see that in the no category, people that don't have overweight people in their family history, the vast majority, 93%, were in the lowest 3 brackets, 72% in normal weight and insufficient weight. This is suggesting that if one of your family members is overweight the chance of you being overweight must increase. The Yes column does support this suggestion, having 56% in the obese categories. It is of note however that in the yes column, more then a quarter of the sample, 28% were in the bottom 3 brackets, showing that is not a perfect correlation and other factors are at play.

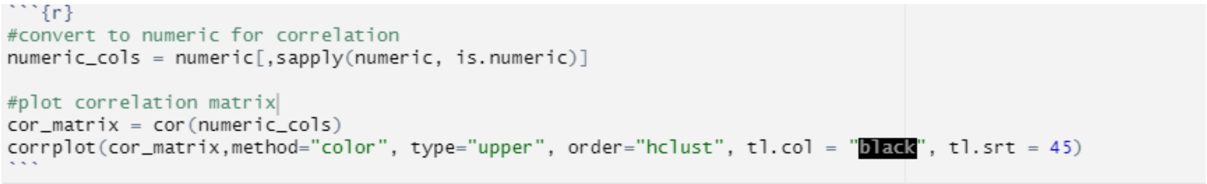
The plot on the right shows the percentage of selections of pre-determined responses to the question Do you eat between meals, from left to right the responses go "no", "Sometimes", "Frequently", "Always". Ranging from least on the left to most on the right. When considering normal weight answerers a clear yet surprising pattern emerges, as with people that do not eat between meals, only 20% of them were normal weight, but for people that always eat between meals 66% of people were normal weight. On the other hand, from people that never eat between meals to people that always do, there was a 15% increase in the amount of people in the obese categories. The column for No show's high bias towards overweight_level_I having 69% whereas the frequently column highlights normal and insufficient weights. Sometimes is unique and shows uniform distribution. Overall, the plot is suggesting a negative correlation between eating between meals and obesity, and that healthier weight people and more likely to eat between meals.



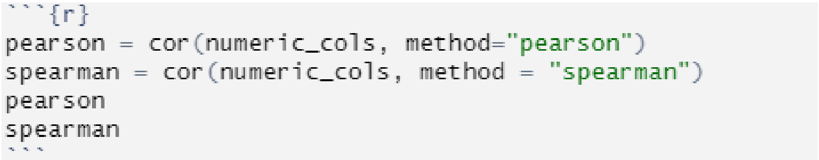
~~~
```{r}
#Load dataset
obesity = read.csv("Obesitydf.csv",header=TRUE)
#remove nas
obesity = na.omit(obesity)
obesity$BMI = obesity$Weight/obesity$Height**2
#show head and structure
head(obesity)
str(obesity)
```
~~~



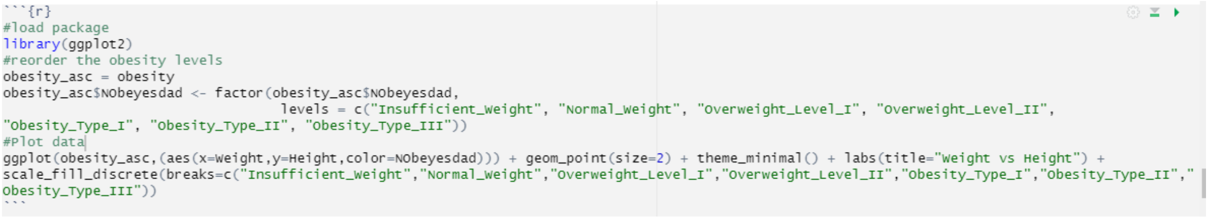
~~~
```{r}
#Analysis of variance
anova_obesity_gender = aov(Age ~ NObeyesdad, data=obesity)
summary(anova_obesity_gender)
anova_obesity_weight = aov(Weight ~ NObeyesdad, data=obesity)
summary(anova_obesity_weight)
anova_obesity_height = aov(Height ~ NObeyesdad, data=obesity)
summary(anova_obesity_height)
```
~~~



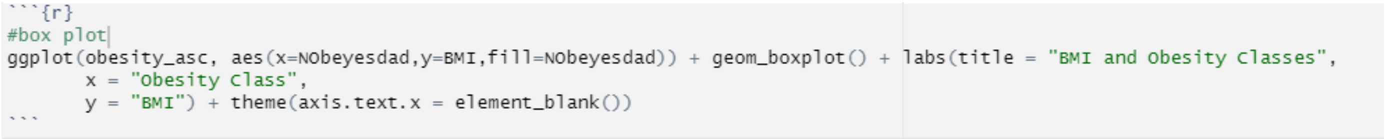
~~~
```{r}
#convert to numeric for correlation
numeric_cols = numeric[,sapply(numeric, is.numeric)]

#plot correlation matrix
cor_matrix = cor(numeric_cols)
corrplot(cor_matrix,method="color", type="upper", order="hclust", tl.col = "black", tl.srt = 45)
```
~~~



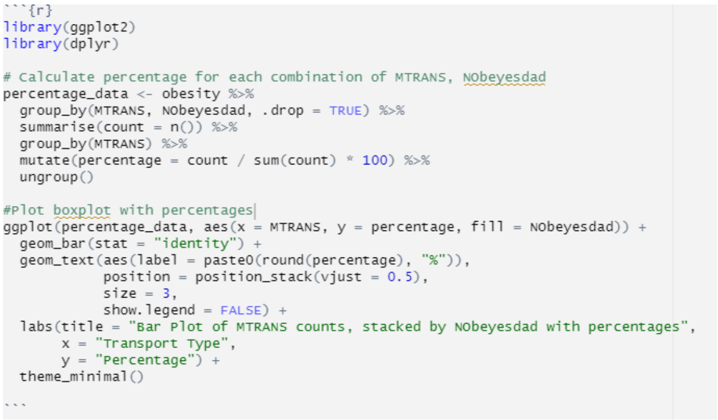
~~~
```{r}
pearson = cor(numeric_cols, method="pearson")
spearman = cor(numeric_cols, method = "spearman")
pearson
spearman
```
~~~



~~~
```{r}
#load package
library(ggplot2)
#reorder the obesity levels
obesity_asc = obesity
obesity_asc$NObeyesdad <- factor(obesity_asc$NObeyesdad,
                                  levels = c("Insufficient_Weight", "Normal_Weight", "Overweight_Level_I", "Overweight_Level_II",
"Obesity_Type_I", "Obesity_Type_II", "Obesity_Type_III"))
#Plot data
ggplot(obesity_asc,(aes(x=Weight,y=Height,color=NObeyesdad))) + geom_point(size=2) + theme_minimal() + labs(title="Weight vs Height") +
scale_fill_discrete(breaks=c("Insufficient_Weight","Normal_Weight","Overweight_Level_I","Overweight_Level_II","Obesity_Type_I","Obesity_Type_II","
Obesity_Type_III"))
```
~~~



~~~
```{r}
#box plot
ggplot(obesity_asc, aes(x=NObeyesdad,y=BMI,fill=NObeyesdad)) + geom_boxplot() + labs(title = "BMI and Obesity Classes",
       x = "Obesity Class",
       y = "BMI") + theme(axis.text.x = element_blank())
```
~~~

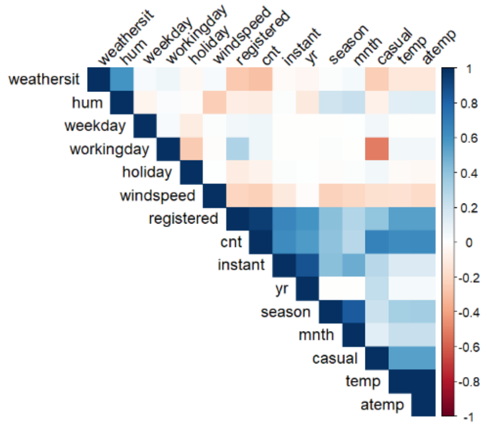


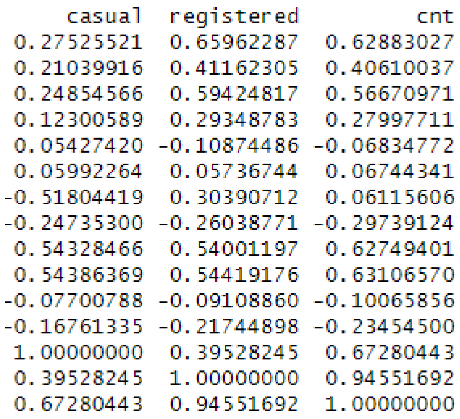
~~~
```{r}
library(ggplot2)
library(dplyr)

# Calculate percentage for each combination of MTRANS, NObeyesdad
percentage_data <- obesity %>%
  group_by(MTRANS, NObeyesdad, .drop = TRUE) %>%
  summarise(count = n()) %>%
  group_by(MTRANS) %>%
  mutate(percentage = count / sum(count) * 100) %>%
  ungroup()

#Plot boxplot with percentages
ggplot(percentage_data, aes(x = MTRANS, y = percentage, fill = NObeyesdad)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = paste0(round(percentage), "%")),
            position = position_stack(vjust = 0.5),
            size = 3,
            show.legend = FALSE) +
  labs(title = "Bar Plot of MTRANS counts, stacked by NObeyesdad with percentages",
       x = "Transport Type",
       y = "Percentage") +
  theme_minimal()

```
~~~

Bikes

The bikes dataset contains the hourly and daily count of rental bikes between the years of 2011 and 2012 in the Capital Bikeshare System with corresponding weather and seasonal information. For our analysis however we will only be looking at the daily use data. People can rent these e-bikes from any location and return to any other place in the city, making them quick, reliable and climate friendly. As they naturally record a lot of data about the journeys, they naturally lend themselves to research project such as this. The dataset has 16 variables with 731 records. Each record has a unique identifier with the instant variable ranging from 1 to 731, and a date staring from January first, 2011, and ending on the 31st of December 2012. Following these are 7 integer variables, representing different characteristics of the specific day in numerical form. For example, we have a column called 'holiday' a binary variable 0 representing not a holiday and 1 representing it was a holiday, or a called 'month' with an integer between 1 and 12 representing the month in which the day took place. Next, we have 4 recorded values of temperature normalised from 8 to 39, another temperature but normalised between 16 and 50, humidity, and windspeed. Followed by 3 count variables, casual or unregistered users, registered users and total, both added together. (https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset)

Again, the to gain a more general understanding of how the variables correlate with each other a Pearson correlation test was undertaken, resulting in this plot and values.

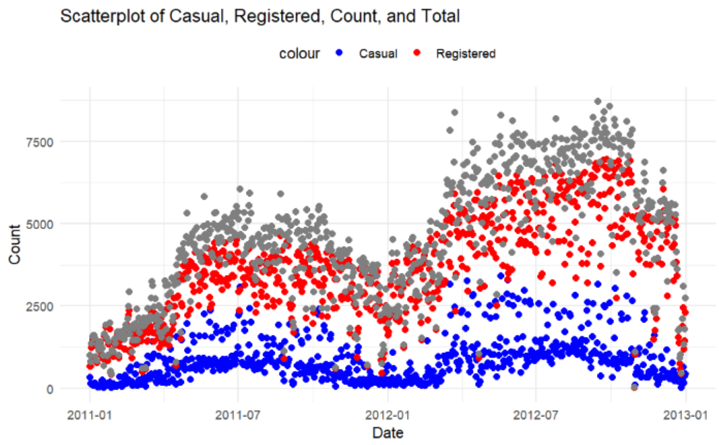




```
      casual  registered         cnt
 0.27525521  0.65962287  0.62883027
 0.21039916  0.41162305  0.40610037
 0.24854566  0.59424817  0.56670971
 0.12300589  0.29348783  0.27997711
 0.05427420 -0.10874486 -0.06834772
 0.05992264  0.05736744  0.06744341
-0.51804419  0.30390712  0.06115606
-0.24735300 -0.26038771 -0.29739124
 0.54328466  0.54001197  0.62749401
 0.54386369  0.54419176  0.63106570
-0.07700788 -0.09108860 -0.10065856
-0.16761335 -0.21744898 -0.23454500
 1.00000000  0.39528245  0.67280443
 0.39528245  1.00000000  0.94551692
 0.67280443  0.94551692  1.00000000
```

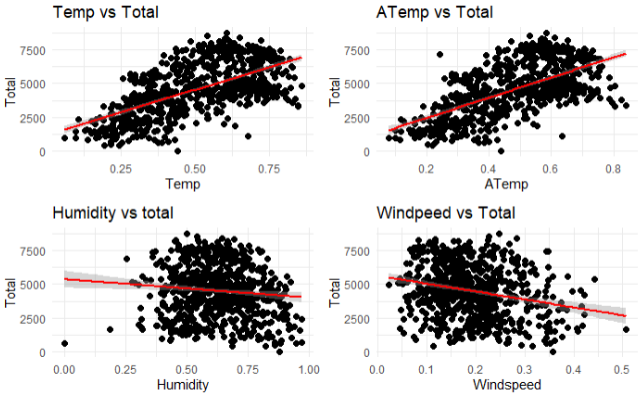
From this we see that the 3 variables that correlated with casual are atemp with 0.543, temp with 0.542 and workingday with -0.518. The 3 variables that correlate with registered are year with 0.594, atemp with 0.544 and temp with 0.540. Finally, the 3 variables that most correlation with the total count cnt is atemp with 0.631, temp with 0.627 and year with 0.567. This is suggesting that overall temp and atemp are the factors that change the response variable the most, with the increase year on year and the decrease in the working day effect it second most. The fact that the workingday variable only highly correlates with is of note, an explanation for this could be that the people using on the weekday are more likely to be using for their trip to work, thus would be regular users and will be more likely to be registered. This is suggesting that throughout the week the demographic of users change, registered users are more likely to be through the week and casual at the weekend.

The best way to visualise the 3 main counts is using a time series graph.



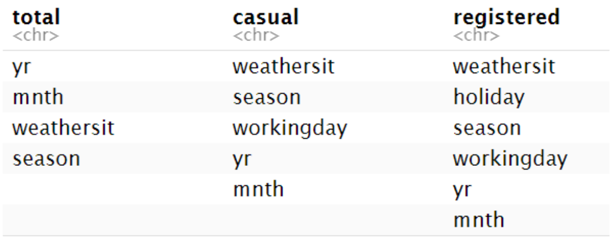
From this we see a yearly harmonic pattern occurring, peaking in the summer months and dipping lower in the winter months, this is expected as it gets colder in those months and people will be less likely to want to use an e-bike. This is also reinforcing the evidence that the two temperature recordings are correlated strong with the amount of people renting the bikes. The other note worthy inference we can make from this graph is the evidence for the year being a positive correlator, as the total counts (labelled in grey) overall dramatically increase from 2011 to 2012, meaning that in the period recorded in this dataset, the overall popularity of the service as increased.

Using scatter plots we can visually see the effect the continuous explanatory variables have on the counts. When doing this we saw that each count had very similar trends, just the counts differed thus I will only show the trends on the total counts.



From this we can observe a strong positive relationship with Temp and ATemp as observed in the correlation test. Humidity seems to have no effect on the total count having slightly negative but near horizontal linear relationship. Finally, a new relationship is discovered, there is a weak negative relationship with windspeed, that becomes much more dramatic and the very large winds. This is a trend understandable to common sense as people could have some fear is riding a bike through high winds.

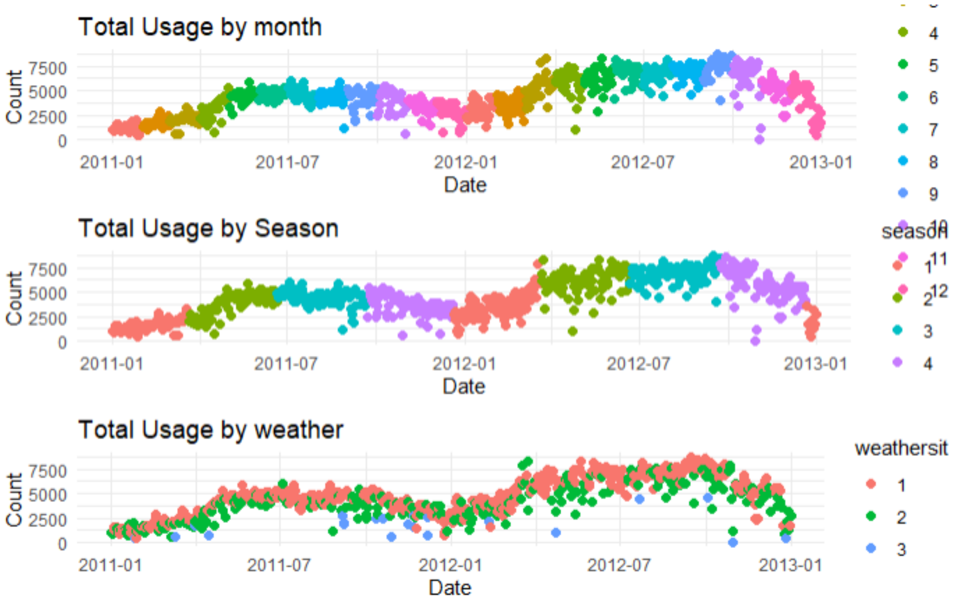
To examine the impact of the categorical variables on the response variables we can complete analysis of variance tests on each variable to gain an insight on if the average between groups differs from the average within the groups. When having a significance level of 0.05 here are the significant variables.

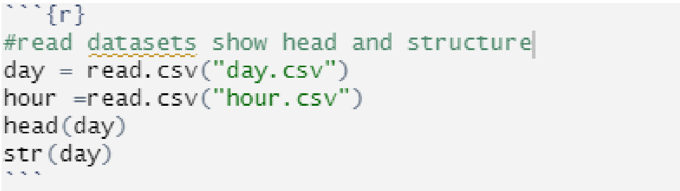


total <chr>	casual <chr>	registered <chr>
yr	weathersit	weathersit
mnth	season	holiday
weathersit	workingday	season
season	yr	workingday
	mnth	yr
		mnth

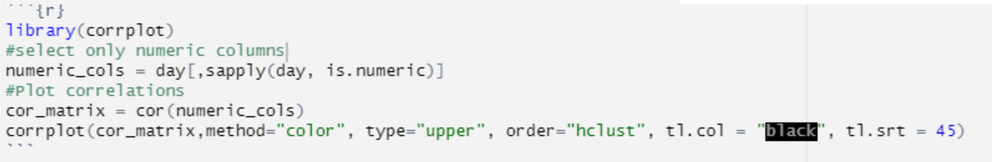
Variables year, month weather and season all effect the number of users in all three count cases, thus we can conclude that they are the most significant variables overall. The number of casual users is also affected by if it's a working day or not, as we previously discovered, this is a negative correlation, as a result it is showing that the number of casual users increases at the weekend. As well as all the previously mentioned variables, registered users are also affected by if the day is a workday holiday, this could be suggesting that a large number of registered users are professionals using the e-bikes on their commute to work.

The time series plots for these significant results are shown below.

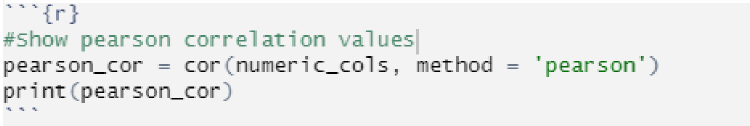




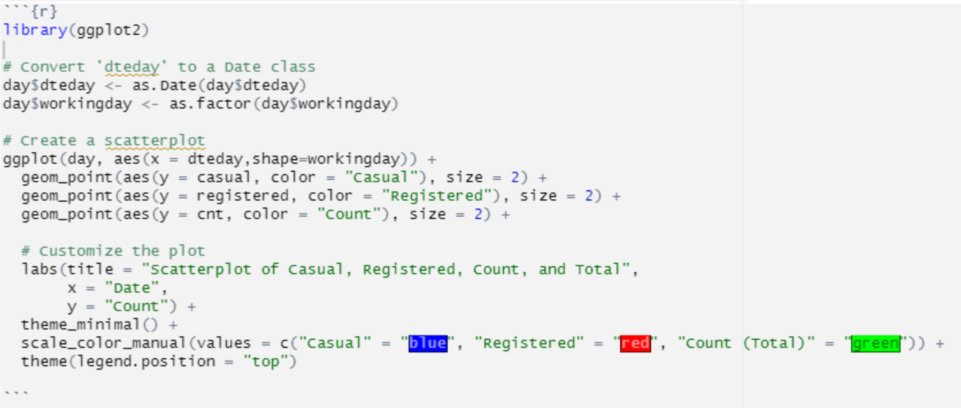
~~~
```{r}
#read datasets show head and structure
day = read.csv("day.csv")
hour =read.csv("hour.csv")
head(day)
str(day)
```
~~~



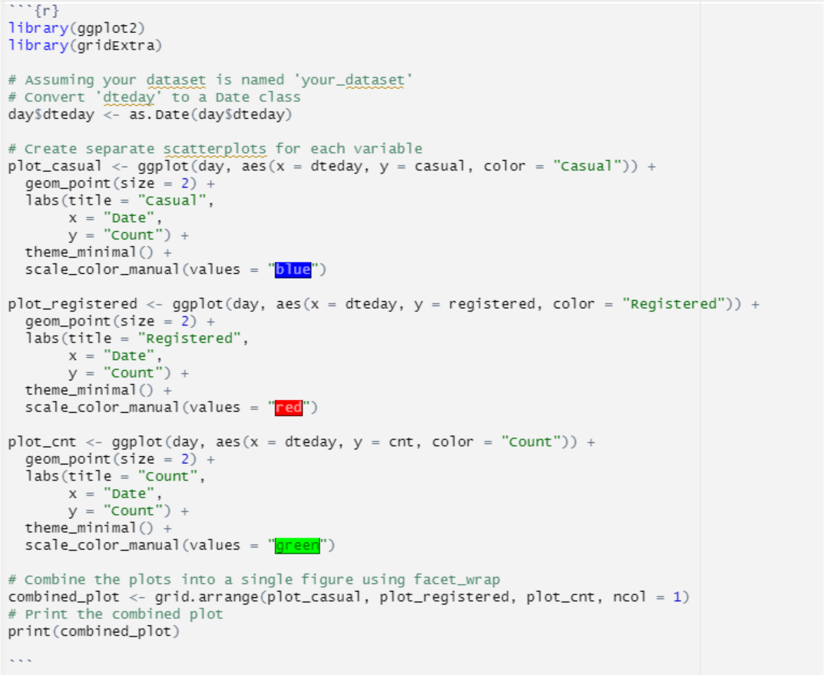
~~~
```{r}
library(corrplot)
#select only numeric columns
numeric_cols = day[,sapply(day, is.numeric)]
#Plot correlations
cor_matrix = cor(numeric_cols)
corrplot(cor_matrix,method="color", type="upper", order="hclust", tl.col = "black", tl.srt = 45)
```
~~~



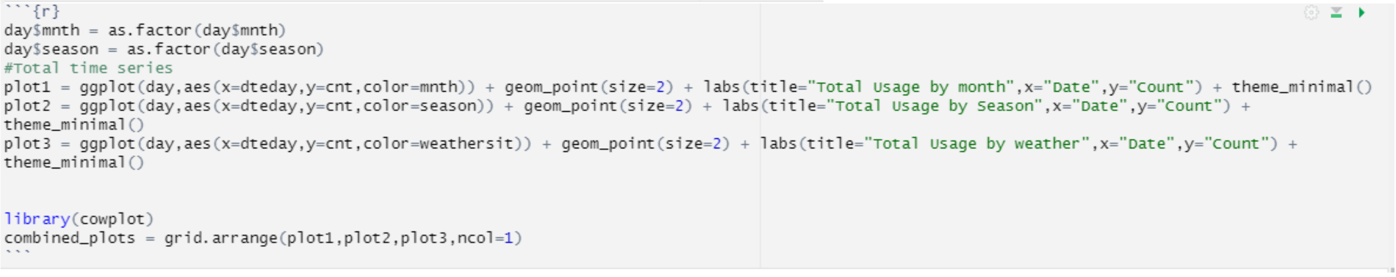
~~~
```{r}
#Show pearson correlation values
pearson_cor = cor(numeric_cols, method = 'pearson')
print(pearson_cor)
```
~~~



~~~
```{r}
library(ggplot2)

# Convert 'dteday' to a Date class
day$dteday <- as.Date(day$dteday)
day$workingday <- as.factor(day$workingday)

# Create a scatterplot
ggplot(day, aes(x = dteday,shape=workingday)) +
  geom_point(aes(y = casual, color = "Casual"), size = 2) +
  geom_point(aes(y = registered, color = "Registered"), size = 2) +
  geom_point(aes(y = cnt, color = "Count"), size = 2) +

  # Customize the plot
  labs(title = "Scatterplot of Casual, Registered, Count, and Total",
       x = "Date",
       y = "Count") +
  theme_minimal() +
  scale_color_manual(values = c("Casual" = "blue", "Registered" = "red", "Count (Total)" = "green")) +
  theme(legend.position = "top")

```
~~~



~~~
```{r}
library(ggplot2)
library(gridExtra)

# Assuming your dataset is named 'your_dataset'
# Convert 'dteday' to a Date class
day$dteday <- as.Date(day$dteday)

# Create separate scatterplots for each variable
plot_casual <- ggplot(day, aes(x = dteday, y = casual, color = "Casual")) +
  geom_point(size = 2) +
  labs(title = "Casual",
       x = "Date",
       y = "Count") +
  theme_minimal() +
  scale_color_manual(values = "blue")

plot_registered <- ggplot(day, aes(x = dteday, y = registered, color = "Registered")) +
  geom_point(size = 2) +
  labs(title = "Registered",
       x = "Date",
       y = "Count") +
  theme_minimal() +
  scale_color_manual(values = "red")

plot_cnt <- ggplot(day, aes(x = dteday, y = cnt, color = "Count")) +
  geom_point(size = 2) +
  labs(title = "Count",
       x = "Date",
       y = "Count") +
  theme_minimal() +
  scale_color_manual(values = "green")

# Combine the plots into a single figure using facet_wrap
combined_plot <- grid.arrange(plot_casual, plot_registered, plot_cnt, ncol = 1)
# Print the combined plot
print(combined_plot)

```
~~~



~~~
```{r}
day$mnth = as.factor(day$mnth)
day$season = as.factor(day$season)
#Total time series
plot1 = ggplot(day,aes(x=dteday,y=cnt,color=mnth)) + geom_point(size=2) + labs(title="Total Usage by month",x="Date",y="Count") + theme_minimal()
plot2 = ggplot(day,aes(x=dteday,y=cnt,color=season)) + geom_point(size=2) + labs(title="Total Usage by Season",x="Date",y="Count") +
theme_minimal()
plot3 = ggplot(day,aes(x=dteday,y=cnt,color=weathersit)) + geom_point(size=2) + labs(title="Total Usage by weather",x="Date",y="Count") +
theme_minimal()


library(cowplot)
combined_plots = grid.arrange(plot1,plot2,plot3,ncol=1)
```
~~~